

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
Anil Singhal, et al.	)	Group Art Unit: 2132
Application No.: 10/637,431	)	Examiner: Jung W. Kim
Filed: August 8, 2003	)	Confirmation No.: 2626
For: INTRUSION DETECTION SYSTEM AND NETWORK FLOW DIRECTOR METHOD	)	

**FILED ELECTRONICALLY**

**Mail Stop RCE**  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

**DECLARATION UNDER 37 C.F.R. § 1.131 ("Declaration A")**

1. I, Danny Lobo, also known as Dionisio Lobo, and I, Anil Singhal, hereby declare:
2. Dionisio Lobo is a named inventor in the above-referenced application ("the Application"). He is also Director of Engineering of assignee NetScout Systems, Inc. of Westford, Massachusetts (NetScout).
3. Anil Singhal is named inventor in the Application. He is also Founder, President, CEO, and Chairman of the Board of assignee NetScout.
4. The Application claims priority to a U.S. provisional patent application, entitled INTRUSION DETECTION SYSTEM AND METHOD, Serial No. 60/402,255 ("the Provisional").

Approved By: RL Date: 2/1/08

5. This declaration is being submitted to establish a conception of the invention disclosed in the Provisional and claimed in the Application prior to July 30, 2002, and to establish diligence in reducing the invention to practice from before July 30, 2002 to August 9, 2002.

**I. EVIDENCE OF CONCEPTION BEFORE JULY 30, 2002**

6. Prior to July 30, 2002, in Westford, MA, we (Danny Lobo and Anil Singhal) conceived the invention disclosed in the Provisional and claimed in the Application. More particularly, prior to July 30, 2002, we conceived of the elements recited in Claim 1, a "method for intrusion detection, comprising: receiving at a probe data packets communicated over a first network link; converting the received data packets into a format suitable for a second network link; monitoring, by the probe, the received packets to evaluate network performance; and transmitting, by the probe over a second network link, data-converted packets to an intrusion detection system in communication with the second network link." Prior to July 30, 2002, in Westford, MA, we also conceived of the elements recited in claims 2; 5-7 and 15, which depend from claim 1 and which recite elements of embodiments of the method for intrusion detection that is recited in claim 1.

7. Prior to July 30, 2002, in Westford, MA, we also conceived of the elements recited in Claim 21, a "network performance probe system comprising: a first network interface for monitoring packets communicated over a first network link; a packet converter for converting the monitored data packets into a format suitable for a second network link; and a second network interface for communicating, over a second network link, converted packets to an intrusion detection system in communication with the second network link." Prior to July 30, 2002, in Westford, MA, we also conceived of the elements recited in claims 24-27, which

depend from claim 21 and which recite elements of embodiments of the network performance probe system that is recited in claim 21.

8. Prior to July 30, 2002, in Westford, MA, we also conceived of the elements recited in Claim 40, "An article of manufacture comprising a program storage medium having computer readable program code embodied therein for providing intrusion detection, the computer readable program code in the article of manufacture including: computer readable code for causing a computer" to perform the method of intrusion detection described above. Prior to July 30, 2002, in Westford, MA, we also conceived of the elements recited in claim 41, which depends from claim 40 and which recites elements of embodiments of the article of manufacture that is recited in claim 40.

9. Prior to July 30, 2002, in Westford, MA, we also conceived of the elements recited in Claim 42, "A program storage medium readable by a computer, embodying a program of Instructions executable by the computer" to perform the method of intrusion detection described above. Prior to July 30, 2002, in Westford, MA, we also conceived of the elements recited in claim 43, which depends from claim 42 and which recites elements of embodiments of the program storage medium that is recited in claim 42.

10. Based on our Invention and prior to July 30, 2002, in Westford MA, we prepared, in the U.S., computer software code, portions of which and printouts of the output for which are attached as Exhibits A - F. These Exhibits demonstrate, through use of a software program time stamp attached here as Exhibit G, conception of the above-described inventions before July 30, 2002.

11. The code that we prepared and that is documented in Exhibits A - H was an implementation of one embodiment of the claimed invention. It comprised a NetScout intrusion detection system operating in conjunction with a NetScout probe model 9200 to detect any

malicious network traffic and network usage which cannot be detected by a conventional firewall.

12. The software programs from which the Exhibits have been extracted contain instructions implementing several functionalities that are neither relevant nor useful to implement the invention claimed in the Application. Only those portions of the software code that are necessary to show the conception of the claimed inventions have been attached as Exhibits hereto; the remainder of the software code has been blocked from its respective Exhibit to preserve its confidentiality. Because the Exhibits are multi-page documents, when an entire page of the Exhibit was blocked, the entire page was deleted from the Exhibit. Also, in accordance with MPEP 715.07.II, dates have been blocked from the Exhibits in order to preserve confidentiality. We declare that the dates are all prior to July 30, 2002.

13. Although other persons contributed to the development of the multiple functionalities implemented in the larger software code, portions of which were extracted and are attached as Exhibits A - H, only Danny Lobo and Anil Singhal prepared the computer software code that implemented the claimed invention as recited in the independent claims.

**A. Method for intrusion detection and a network performance probe**

14. Exhibit A consists of portions of a software program entitled DRVCFG.C, which contains instructions to implement a "method for intrusion detection." Page 1 of Exhibit A shows that DRVCFG.C bears a copyright notice in the name of Frontier Software Dev. Inc., which was the name of NetScout at the time that the software program DRVCFG.C was created. DRVCFG.C identifies the software's subject matter as the probe Model 9200 and the intrusion detection system known in the program as "IDS". See, for example, Code Fragment 1 of Exhibit A (page 3): "#elif (MODEL\_9200 && \_IDS)."

**B. Communicating data packets over a first network link and receiving them at the probe**

15. As shown in Exhibits A - C, the instructions include "receiving at a probe data packets communicated over a first network link." DRVCFG.C has instructions that comprise a call to detect a wide area network (WAN) link (called here "CC3i") from the driver call table. See, for example, Code Fragment 2 of Exhibit A (page 7):

```
"#if CC3i
    if (detect_cc3i(ifn))
    {
        log_event("Interface %d: (HSSI) WAN-CC3i\n", ifn);
        return(&cc3i_wan_info);
    }
#endif"
```

The WAN link CC3i is an embodiment of a first network link, such as recited in independent Claims 1, 21, 40, and 42.

16. The software program entitled CC3I.API.C, a portion of which is attached as Exhibit B, contains routines to detect, initialize and receive packets through the WAN link CC3i. CC3I.API.C includes instructions for receiving a data frame over the first WAN link CC3i. See, for example, Code Fragment 1 of Exhibit B, page 16 ("EXPORT XMIB\_FRAME\_INFO \*cc3i\_recv\_frame(ifn)").

17. CC3I.API.C has Instructions for detecting the WAN link CC3i. See, for example, Code Fragment 2 of Exhibit B (pages 21 - 22):

```
"EXPORT BOOL detect_cc3i(ifn)
UINT ifn;
{

    memset((void *) cc3i, 0, sizeof(CC3I_INFO));

    /*
    ** Locate CC3i card using PCI find device function, read its
    ** configuration and setup the cc3i h/w access structure.
    */
    if ( !locate_cc3i(CC3I_VENDOR_ID, CC3I_DEVICE_ID, cc3i) )
```

```
{
log_event("cc3i: failed to locate device\n");
return(FALSE);
}"

cc3i->ifn = ifn;

// Set CRC16 as default option
// -----
// agent_config.if_data[ifn-1].options |= CFG_MASK(IFOPT_CRC_16);

return (cc3i_detected = TRUE);
}"
```

18. The software program entitled RTPROC.C, a portion of which is attached as Exhibit C, contains instructions to provide a conduit between the WAN link CC3i, to receive packets, and to pass them along to the network performance evaluation code portion of the probe. RTPROC.C has instructions to communicate data packets over the WAN link CC3i. See, for example, Code Fragment 1 of Exhibit C (page 3):

```
"xmlb_frame_info = drv_rcv_frame(ifn);
if (xmlb_frame_info == 0)"
```

19. RTPROC.C has instructions to send packets along the WAN link CC3i to the network performance evaluation code portion of the probe. See, for example, Code Fragment 2 of Exhibit C (page 4):

```
"if (!collector_count)
    mibmgr_collector(xmlb_frame_info);
else{
    /*
    ** Test case to simulate
    ** multiple frames.
    */
    for (j = 0; j < collector_count; j++)
        mibmgr_collector(xmlb_frame_info);"
```

C. **Monitoring the received packets by a probe to evaluate network performance**

20. As shown in Exhibit D, the instructions include "monitoring, by the probe, the received packets to evaluate network performance." For example, the software program

entitled REALCOL.C, a portion of which is attached as Exhibit D, contains instructions to provide network performance evaluation. Performance is evaluated through collection of data in Management Information Bases (MIBs). An MIB is a type of database used to manage the devices in a communications network. It comprises a collection of objects in a (virtual) database used to manage entities (such as routers and switches) in a network.

21. REALCOL.C contains instructions that operate to evaluate performance in MIBs.

See, for example, Code Fragment 1 of Exhibit D (page 6):

```
"EXPORT VOID mibmgr_collector(info)
    XMIB_FRAME_INFO *info;"
```

22. See also Code Fragment 2 of Exhibit D (page 6):

```
"TRACE(MIBMGR_COLLECTOR);
if (info->channel)
{
    /* setup channel interface indexes */
    info->ifn = info->channel->ch_ifn;
    info->lifn = info->channel->ch_lifn;
}

fastpath_process_frame(info);"
```

**D. Converting the received data packets by a packet converter into a second network link format**

23. As shown in Exhibits A, D – F, the instructions include "converting the received data packets into a format suitable for a second network link."

24. As noted above in paragraph 14, the software program entitled DRVCFG.C (Exhibit A) contains instructions to provide the method for intrusion detection with two network links. DRVCFG.C identifies the driver "FEC" and the driver "CC3i." See, for example, Code Fragment 1 of Exhibit A (page 4):

```
"#define FEC 1
#define CC3i 1"
```

25. As noted above in paragraph 11 above, the driver "CC3i" constitutes a wide area network (WAN) link. The WAN link CC3i is an embodiment of the first network link, such as recited in the claims.

26. The software program entitled FECMAIN.C, a portion of which is shown as Exhibit E, is identified as an Ethernet driver. For example, the comments on page 1 of Exhibit E:

```
"Module Name: FecMain.c  
Component of: Fast Ethernet driver using the Adaptec Starfire adapter"
```

The Ethernet driver FECMAIN.C is an embodiment of a second network link, such as recited in independent Claims 1, 21, 40, and 42.

27. In the system implementation demonstrated by Exhibits A – F, after transmission along the WAN link CC3i, the data packets are converted by packet converter code and then are forwarded to a software program called "fecsendframe()" in the format suitable for the Ethernet link FECMAIN.C. FECMAIN.C has instructions to receive the converted frames from FECSENDERFRAME.C. See, for example, Code Fragment 2 of Exhibit E (page 47):

```
"EXPORT INT FecSendFrame (ifn, frameptr, framesize, TrafGenMode)  
  
    UINT  ifn;  
    N8    *frameptr;  
    UINT  framesize;  
    N32   TrafGenMode;  
  
    {  
        SF_ADAPTER      *Adapter;    // Pointer to Adapter structure  
        XMIB_FRAME_INFO *info;"
```

28. Exhibit F is a printout of the output of a software program entitled F9200, which identifies the topology and applications supported by the NetScout probe model 9200 in the system implementation evidenced by Exhibits A – F. It can be seen in Exhibit F that the system implementation contains a WAN link ("-D\_WAN=1") and an Ethernet link ("-D\_ET=1"). Thus, Exhibit F shows that the data packets will transmit along two different network links.



29. Persons skilled in implementing network applications will understand that that the format of data transmitted along a WAN link will differ from the format of data transmitted along an Ethernet link. In reviewing the system implementation evidenced by Exhibits A – F, a person skilled in implementing network applications would understand that data packets that were first transmitted through a WAN network link require conversion into a different format before they could be transmitted along an Ethernet link. Therefore, a person skilled in implementing network applications would understand that the system implementation evidenced by Exhibits A – F would require conversion of the format of the received data packets from the format suitable for the first network link into a format suitable for the second network link. Thus, in our opinion, the computer software code and output of Exhibits D – F show that the received data packets are converted from a format suitable for a first network link into a format suitable for a second network link

**E. Transmitting data-converted packets over a second network link to an intrusion detection system in communication with the second network link**

30. As shown in Exhibit A, E, F, the instructions include “transmitting, by the probe over a second network link, data-converted packets to an intrusion detection system in communication with the second network link.”

31. Both Exhibit A (DRVCFG.C) and Exhibit F (F9200) demonstrate that the system implementation contains an intrusion detection system. See, for example, Code Fragment 1 of Exhibit A (page 3) (“#elif (MODEL\_9200 && \_IDS)”) and the last line of Exhibit F (“-D\_IDS=1”).

32. FECMAIN.C (Exhibit E) is invoked in the system implementation to transmit and forward the data-converted packets to the intrusion detection system in communication with it. FECMAIN.C has a call routine to detect the Ethernet link. See, for example, Code Fragment 1 of Exhibit E (pages 22 – 23), :

```
"EXPORT TEXT *FecDetect(ifn)
  UINT ifn;"
```

33. Further, FECMAIN.C has instructions for sending the converted frames over the Ethernet link to the intrusion detection system. See, for example, Code Fragment 2 of FECMAIN.C (Exhibit E, page 47):

```
"CHK_IFN(ifn);
.
.
.

Adapter = InterfaceInfo[ifn].Adapter[0];

// First check if a previously queued frame needs to be ACKed
AckPrevXmtdFrame (Adapter);

// Q the frame for xmission
if (!XmtFrame (Adapter, frameptr, framesize,
  (TrafGenMode == CRC_ERR) ? FALSE:TRUE))
{
    //
    // Update Stats for this frame if it was successfully Qued.
    //
    info = Adapter->RecvFrameInfo;
    info->frame_time_ms = timer_get_uptime();
    info->frame_p = frameptr;
    info->frame_size = framesize;
    info->crc_align_error = (TrafGenMode == CRC_ERR) ? TRUE :
        FALSE;
    info->mac_errors = info->crc_align_error;
    Adapter->TxmLastBuffer = (N8 *)frameptr;
    Adapter->TxmPrevPend = TRUE;"
```

**F. Software run time-stamp**

34. Exhibit G is a printout of the output of a software program entitled LINKTIME.C, which operates as a time-stamp, evidencing the execution of the software of Exhibits A - F prior to July 30, 2002. The date itself has been blocked to preserve its confidentiality in accordance with MPEP 715.07.II, but we declare that the time-stamp date was prior to July 30, 2002.

35. The probe and intrusion detection method disclosed in the computer software code shown in Exhibits A - F form the subject of the above-identified application. As can be seen from the above, substantially all of the features set forth in the independent claims of the above-identified application are also described in the computer software code and printout of the system implementation shown in Exhibits A - F. In our opinion, the computer software code shown in Exhibits A - E and the printout of the system implementation shown in Exhibit F, which, as evidenced by the printout of the output of the entitled LINKTIME.C program shown in Exhibit F, are dated prior to July 30, 2002, conclusively demonstrate the conception of the claimed invention prior to July 30, 2002 and acts supporting such conception in the U.S.

## **II. DILIGENCE FROM JULY 30, 2002 TO AUGUST 9, 2002**

36. Prior to July 30, 2002, in Westford, MA, a patent application drafting project was started for this invention. The project resulted in the filing of the Provisional on August 9, 2002. The law firm handling patent matters on behalf of NetScout assigned Docket No. FSD-004 to the Provisional. On and after July 30, 2002 and prior to August 9, 2002 (the filing date for the Provisional), a draft of the Provisional was being revised.

37. As evidence of such work, attached hereto as Exhibit H is a copy of a portion of the law firm's invoice to assignee NetScout showing the attorney hours spent on reviewing and revising the draft Provisional. As evidenced by the attached copy of selected pages of the invoice, on July 25 and 26, 2002, 11.50 hours of attorney time were spent on matters on behalf of the assignee NetScout including a review and revision of the draft application for FSD-004. Thereafter, 4.25 hours of attorney time were spent on July 30, 2002 reviewing correspondence from Danny Lobo and conferring with Danny Lobo about the draft Provisional. Further, on Friday, August 2, 2002, 0.50 hours of attorney time were spent reviewing a document from Danny Lobo. After a break for the weekend, August 3 - 4, 2002, on Monday, August 5, 2002, an

additional 4.00 hours of attorney time were spent reviewing Danny Lobo's document and revising the application. On August 6, 2002, 0.50 hours of attorney time were spent reviewing the application and, on August 7, 2002, 2.00 additional hours of attorney time were spent reviewing and revising the draft Provisional, which was filed two days later.

38. Clearly, the hours that Danny Lobo spent in reviewing versions of the draft Provisional in that period would be in addition to those hours of attorney time that were documented in Exhibits H.

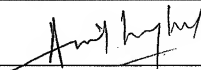
39. In our opinion, the copy of a portion of the law firm's invoice to NetScout that is attached hereto as Exhibit H and that shows significant work performed in drafting the provisional patent application upon which the instant application is based in the period between July 30, 2002 and August 9, 2002 conclusively demonstrates that we were diligent in achieving reduction to practice of the invention after July 30, 2002 but before August 9, 2002.

40. We declare further that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further, that the statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patents issuing thereon.

Dated: 2/1/08



Dated: 2/5/08



## Exhibit A

drvcfg

```

/*****
*      COPYRIGHT (c)      , FRONTIER SOFTWARE DEV. INC
*      ALL RIGHTS RESERVED
*
* Module Name : drvcfg.c
* Component of: MPC Software
* Programmer  : Anil Singhal
*
* Description:
*      Contains driver configuration info for all models.
*      Must be compiled using appropriate MODEL_xxxx switch.
*
* Comments:
*
* Revision History:
*
* Vers. Date      who why
*
*
*
*
*****/
```

## Exhibit A

drvcfg

```
#elif (MODEL_9200 && _IDS)
#define FEC      1
#define CC3i    1
```

## Exhibit A

drvcfg

```
#if CC3i
    if (detect_cc3i(ifn))
    {
        log_event("Interface %d: (HSSI) WAN-CC3i\n", ifn);
        return(&cc3i_wan_info);
    }
#endif
```

## Exhibit B

```
                                cc3iapi
#define USE_RESERVED_MEMORY    1
/* *****
/*      Copyright (c)          , Frontier Software Development, Inc.
/*                               ALL RIGHTS RESERVED.
/*
/*      Module Name:          cc3iapi.c
/*      Component of:         CC3i, HSSI driver API library.
/*      Programmer:           Rajeev Nadkarni.
/*
/*      Description:
/*
/*      Contains all functions pertaining to the HSSI device driver using
/*      SDL Communications CC3i card.
/*
/*      Comments:      Use DO_TEST switch while testing code on DOS platform.
/*                      Always use interface number (ifn) 1 in Test mode
/*
/*      Revision History:
/*
/*      Vers.  Date          who          why
/*
/* *****
```



## Exhibit B

cc3iapi

```
EXPORT XMIB_FRAME_INFO *cc3i_recv_frame(ifn)
```

```
*,  
*
```

## Exhibit B

cc3iapi

```
EXPORT BOOL detect_cc3i(ifn)
    UINT ifn;
    {
        memset((void *) cc3i, 0, sizeof(CC3I_INFO));

        /*
        ** Locate CC3i card using PCI find device function, read its
        ** configuration and setup the cc3i h/w access structure.
        */
        if ( !locate_cc3i(CC3I_VENDOR_ID, CC3I_DEVICE_ID, cc3i) )
            {
                log_event("cc3i: failed to locate device\n");
            }
    }
    }
```

## Exhibit B

```
        return(FALSE);          cc3iapi
    }

    cc3i->ifn = ifn;

    // Set CRC16 as default option
    // -----
    // agent_config.if_data[ifn-1].options |= CFG_MASK(IFOPT_CRC_16);
    return (cc3i_detected = TRUE);
}
```

## Exhibit C

```
rtproc

#if MODEL_9300
#include rtproc_a.c
#else
/*****
*
*      W A R N I N G
*
*      Copyright (c)
*      NetScout Systems, Inc.
*      Westford, MA 01886
*      All Rights Reserved
*
* This software is part of Licensed material, which is the property of
* NetScout Systems, Inc. Unauthorized use, duplication or distribution
* is strictly prohibited by Federal law. No title to and ownership of
* this software is hereby transferred.
*
*****/
/*****
*
* Module Name : rtproc.c
* Component of: MPC Software
* Programmer  : Anil Singhal
*
* Description:
*              Contains the top level real time procedure.
*
* Comments:
*
* Revision History:
*
* Vers. Date      Who      Why
*
*
*
*
*
*
*
*
*
*
*
*
*****/
*****/
```

## Exhibit C

rtproc

```
xmib_frame_info = drv_rcv_frame(ifn);  
if (xmib_frame_info == 0)
```

## Exhibit C

rtproc

j++)

```
if (!collector_count)
    mibmgr_collector(xmib_frame_info);
else{
    /*
    ** Test case to simulate
    ** multiple frames.
    */
    for (j = 0; j < collector_count;
        mibmgr_collector(xmib_frame_info);
```

## Exhibit D

```

                                realcol
#define RMON1_CHANGES_FOR_RMON2 1
#define MCAST_BCAST_FIX 1
/*****
*                               W A R N I N G
*                               ~~~~~
*                               Copyright (c)
*                               NetScout Systems, Inc.
*                               Westford, MA 01886
*                               All Rights Reserved
*
* This software is part of Licensed material, which is the property of
* NetScout Systems, Inc. Unauthorized use, duplication or distribution
* is strictly prohibited by Federal law. No title to and ownership of
* this software is hereby transferred.
*
*****/
/*****
*
* File Name : realcol.c
* Component of: MIB Manager
* Programmer : Anil Singhal
*
* Description:
*   Contains all top-level Collector functions for LICs.
*   Includes all channel and filter related functions also
*
* Comments:
*
* Revision History:
*
* Vers. Date      who why
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*****/
/
```

## Exhibit D

```
realcol

EXPORT VOID mibmgr_collector(info)
    XMIB_FRAME_INFO *info;

    TRACE(MIBMGR_COLLECTOR);
    if (info->channel)
    {
        /* setup channel interface indexes */
        info->ifn = info->channel->ch_ifn;
        info->lifn = info->channel->ch_lifn;
    }
    fastpath_process_frame(info);
```



## fecmain

Page 1

## Exhibit E

fecmain

## Exhibit E

```
UINT ifn;
fecmain
```

## Exhibit E

fecmain

```
EXPORT INT FecSendFrame (ifn, frameptr, framesize, TrafGenMode)
    UINT    ifn;
    N8      *frameptr;
    UINT    framesize;
    N32     TrafGenMode;
{
    SF_ADAPTER *Adapter;           // Pointer to Adapter structure
    XMIB_FRAME_INFO *info;

    CHK_IFN(ifn);

    // Make sure that a FDX interface is not used for Xmt
    if (InterfaceInfo[ifn].MaxBoards != 1)
    {
        log_event (
            "<FecSendFrame>: Xmt not supported on FDX Interfaces [Current Ifn %d]",
            ifn);
        return (EDRV_SUCCESS);
    }

    Adapter = InterfaceInfo[ifn].Adapter[0];

    // First check if a previously queued frame needs to be ACKed
    AckPrevXmtedFrame (Adapter);

    // Q the frame for xmission
    if (XmtFrame (Adapter, frameptr, framesize,
        (TrafGenMode == CRC_ERR) ? FALSE:TRUE))
    {
        //
        // Update Stats for this frame if it was successfully queued.
        //
        info = Adapter->RecvFrameInfo;
        info->frame_time_ms = timer_get_uptime();
        info->frame_p = frameptr;
        info->frame_size = framesize;
        info->crc_align_error = (TrafGenMode == CRC_ERR) ? TRUE : FALSE;
        info->mac_errors = info->crc_align_error;
        Adapter->TxmLastBuffer = (N8 *)frameptr;
        Adapter->TxmPrevPend = TRUE;
    }
}
```

## Exhibit F

f9200

-D\_ET=1  
-D\_ETM=1  
-D\_FET=1  
-D\_FETM=1  
-D\_WAN=1  
-D\_RM=1  
-D\_NF=1  
-D\_CF=1  
-D\_CDP=1  
-DNEW\_NETFLOW=1  
-DUSR\_HISTORY=1  
-DSMON\_SUPPORT=1  
-DHCRMON\_SUPPORT=1  
-DPROBE\_CONFIG\_BACKEND=1  
-DFULL\_PROBE\_CONFIG=1  
-DCPU\_DETECTION=1  
-DTOPN\_NL\_MATRIX=1  
-DTOPN\_AL\_MATRIX=1  
-DTOPNHOST\_SUPPORTED=1  
-DFASTPATH=1  
-DARTMIB\_SUPPORT=1  
-DRMON\_FILTER=1  
-D\_DSMON=1  
-D\_HTTPMIB=1  
-D\_FEC=1  
-DARUBA\_BOARD=1  
-DART\_ON\_IFN49=1  
-DENTERPRISE\_PATCH=1  
-D\_LINKAGGR=1  
-DPERF\_ARUBA=1  
-D\_VOIP=1  
-D\_IDS=1

## Exhibit G

```
char nsagent_linktime[] = "linktime.c";
```

Netscout Systems, Inc.

Invoice: 197095

Page 13

07/25/2002	[REDACTED]	FSD-004: review and revise draft; confer Martinson regarding draft	3.50	[REDACTED] Pat
07/25/2002	[REDACTED]	(FSD-004) Incorporated the comments from Mr. Lobel into the application. Updated the application and figures with Mr. Heffan comments. Drafted email to Mr. Lobel re: publication.	4.25	[REDACTED] Pat
07/26/2002	[REDACTED]	FSD-004: Review and revise application.	1.25	[REDACTED] Pat
	[REDACTED]	[REDACTED]		[REDACTED] Pat
07/26/2002	[REDACTED]	(FSD-004) Reviewed agent manual disclosure. Incorporated Mr. Frank's comments into the application. Office conference with Mr. Heffan re: application. Drafted email to Mr. Lobel re: draft application.	2.50	[REDACTED] Pat
	[REDACTED]	[REDACTED]		[REDACTED]
	[REDACTED]	[REDACTED]		[REDACTED]
	[REDACTED]	[REDACTED]		[REDACTED]
07/30/2002	[REDACTED]	FSD-004: review correspondence from Lobo; telephone conferences with Lobo to review application draft	1.50	[REDACTED] Pat
	[REDACTED]	[REDACTED]		[REDACTED]
	[REDACTED]	[REDACTED]		[REDACTED]
	[REDACTED]	[REDACTED]		[REDACTED]
	[REDACTED]	[REDACTED]		[REDACTED]
	[REDACTED]	[REDACTED]		[REDACTED]
	[REDACTED]	[REDACTED]		[REDACTED]
	[REDACTED]	[REDACTED]		[REDACTED]
07/30/2002	[REDACTED]	(FSD-004) Telephones conference with Mr. Heffan and Mr. Lobel re: draft	2.75	[REDACTED] Pat

# Exhibit H

TEA, HURWITZ & THIBEAULT, LLP

NetScout Systems, Inc.

Invoice: 198190

Page 23

FSD-004 Enhanced Computer Network Intrus *IR*

Date	Name	Description	Hours	Amount
08/02/2002	[REDACTED]	FSD-004: review document from Lobo	0.50	[REDACTED]

08/05/2002	[REDACTED]	FSD-004: review draft document from Lobo; review and revise patent application	4.00	[REDACTED]
------------	------------	--	------	------------

08/08/2002	[REDACTED]	FSD-004: review changes to application with Martinson	0.50	[REDACTED]
------------	------------	---	------	------------

08/07/2002	[REDACTED]	FSD-004: review and revise patent application per Lobo comments; review changes with Martinson	2.00	[REDACTED]
------------	------------	--	------	------------